Autonomous Car for Indian Conditions

Riya Manoj, Anand C A, Muhamed Abdul Rehman, Ansu Mars Puthettu, Dr.Sherly K.K
Information Technology, Rajagiri School of Engineering & Technology
Assoc. Professor, Information Technology, Rajagiri School of Engineering & Technology

Abstract — Self-driving cars are vehicles that perceive their surroundings and manoeuvre without the assistance of a driver. This is a high-research area in computer vision that covers many sub-topics and requires in-depth analysis. In order to do this, our study examines the hardware and software elements of a self-driving automobile, utilising Deep Learning technologies such as Convolution Neural Networks, the YOLO algorithm, Hough Transform Algorithms and the Canny Edge Detection algorithm. The modules used are perception, localization, mapping, vehicle path planning, vehicle control. Thispaper reasons some of the problems in the existing technology and provides a few solutions that can be taken to handle it.

Keywords— Convolution neural networks, self-driving car,object detection, traffic sign detection, lane detection.

I. INTRODUCTION

Number of vehicles on the road has increased over the years and with this growth the number of accidents has also increased. Most accidents are caused by human inattention or a slow response time. Disregarding traffic rules and human error may contribute to accidents and serious injuries to not only themselves but other motorists, passengers, pedestrians etc. An autonomous car can be defined as a vehicle which is able to perceive its environment, decide which route to take to its destination, and drive it. It uses object detection algorithms such as YOLO to accurately determine objects. YOLO (You Only Look Once) is a type of convolutional neural network (CNN) that is commonly used for object detection in images and video. It is intended to have fast inference speed and ability to process images in real-time.

In the context of autonomous vehicles, YOLO could be used to detect and classify objects in the environment, such as other vehicles, pedestrians, traffic signs and barriers in the vehicle's vicinity. This information is then used by the autonomous vehicle's decision-making system to navigate and make driving decisions. Cars equipped with this technology will likely reduce crashes, energy consumption, and considerably pollution. Proposed system is intended to detect and recognize human beings, vehicles, traffic lights, potholes and other road objects in real time and drive without human intervention and also helps in providing greater road safety, greater independence, more productivity, reduced congestion, environmental gains. By the introduction of autonomous vehicles, accidents will be significantly reduced. Driverless cars could make mobility more accessible for those currently unable to drive. It may allow seniorcitizens, people with disabilities and potentially even children, greater access to independent commuting. Autonomous vehicles will also likely benefit the economy through fuel efficiency, the environment through reduced carbon emissions.

II.PROBLEM DEFINITION

Most road accidents are caused by human inattention or a slow response time. The conventional object detection in autonomous vehicles is less efficient, less accurate in detecting objects at night, small or nearby objects and costly. Developing an autonomous car for Indian conditions involves addressing challenges such as unstructured traffic, poor road infrastructure, pedestrian safety, extreme weather conditions. Most road accidents are caused by human inattention or a slow response time. The conventional object detection in autonomous vehiclesis less efficient, less accurate in detecting objects at night, small or nearby objects and costly. Developing an autonomous car for Indian conditions involves addressing challenges such as unstructured traffic, poor road infrastructure, pedestrian safety, extreme

The problem at hand is the development of a self-driving car system that can accurately distinguish between stationary and moving objects such as vehicles, animals, and other objects onthe road while maintaining safe driving conditions. Additionally, the system must be capable of detecting these objects from a significant distance in order to ensure that the car can react appropriately, whether it requires a change in speed, direction or the application of brakes. This presents a complex challenge as it involves the use of advanced algorithms and sensors to accuratelydetect and recognize objects in a dynamic and ever-changing environment, while ensuring the safety of both the vehicle's occupants and those around it.

III.PROPOSED SOLUTION

A proposed solution for developing an autonomous car for Indian conditions could involve incorporating advanced sensors and AI algorithms that can handle the unpredictable traffic, poor road infrastructure, and extreme weather conditions. The proposed system contains four modules, the first part is the perception module. This module is responsible for detecting the surrounding environment of the vehicle using sensors such as cameras, sonar and sensors. It involves tasks such as object detection, recognition, and tracking. It uses YOLOv3 object detection algorithm to accurately determine objects such as vehicles, potholes, traffic signs, pedestrians, and barriers in the vehicle's vicinity.
Second part is localization; this module determines the precise position of the vehicle in its environment. It uses GPS NEO6M, IMU, compass sensors, to estimate the vehicle's location and bearing. Next is mapping; This module builds a map of the environment around the vehicle using data from the perception and localization modules. It includes tasks such as creating a high-definition map of the road and identifying lane markings, traffic signs, and traffic lights. For accurately detecting the lane boundaries this proposal uses Canny edge detection algorithm. Last part is path planning; this module uses information from the perception, localization, and mapping modules to plan a safe and efficient route for the vehicle. It takes into account factors such as traffic regulations, road conditions such as potholes and speed bumps, and obstacles in the environment. It uses A* algorithm to decide the travel path. Hence the vehicle is able to perceive its environment, decide which route to take to its destination, and drive it safely. The proposed system architecture is shown in figure 1.

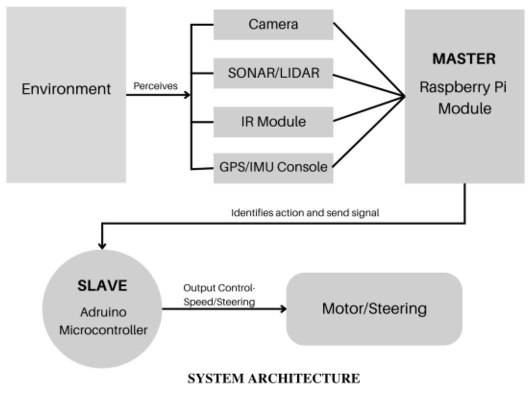


Fig .1. Proposed System Architecture

IV.HARDWARE AND SOFTWARE REQUIREMENTS

Hardware components to construct the self-driving automobile prototype are Raspberry Pi 3, Pi Camera, and Arduino. A self-driving car's pi camera, which is mounted on the top, uses live images to take pictures that are then sent to the Raspberry Pi for image processing, namely convolution neural networks. By using a USB cable to connect to a computer, the Arduino board connects with Arduino, which then uses a remote controller to deliver signals to a motor in accordance with the output of the neural network. Car moves in accordance with the signal as it is activated.

1. Raspberry pi 3

A key element in many initiatives, including the development of self-driving car prototypes, is the Raspberry Pi 3. It is a little single-board computer. Along with Wi-Fi and Bluetooth built-in,it has a quad-core ARM Cortex-A53 CPU. The Raspberry Pi 3 has the processing power and flexibility to execute complicated algorithms, manage sensor data, and manage many areas of the self-driving car's operation.The Raspberry Pi 3 acts as the central processing unit (CPU) of a self-driving car prototype, carrying out operations including image processing, sensor fusion, decision-making algorithms, and interfacing with other parts. Tosense the surroundings and acquire pertinent information for autonomous navigation, it communicates with a variety of sensors, including cameras, radar, and lidar.

2. Arduino UNO

The Arduino Uno is a microcontroller board that incorporates the ATmega328P microcontroller at its core It has 32KB of flash memory, of which 0.5KB is set aside for the bootloader, and runsat a clock rate of 16 MHz. For data storage, the board provides 2KB of SRAM and 1KB of EEPROM. The Arduino Uno offers14 digital input/output pins, 6 of which can be used as PWM (Pulse Width Modulation) outputs, for input and output functionality. It also contains six analogue input pins. Through its USB interface, the Uno supports serial connection, making it simple to upload code and communicate with a computer. Additionally, it has dedicated SPI (Serial Peripheral Interface) and I2C (Inter-Integrated Circuit) communication pins for establishing connections with other compatible devices. The USB connection or an external power source can be used to power the board.[1]

Pi camera

A small camera created especially for use with Raspberry Pi boards is called the Pi Camera, sometimes known as the Raspberry Pi Camera Module. A ribbon wire connects the Pi Camera's compact form-factor camera sensor to the Raspberry Pi's camera connector. The current Pi Camera variants come with either an 8-megapixel Sony IMX219 sensor or a 12.3-megapixel Sony IMX477 sensor. There are various versions of the Pi Camera. The Pi Camera can record films and photos in high definition. Still photos with a maximum resolution of 3280 x 2464 pixels are supported by the 8-megapixel version. There are several resolutions at which video can be recorded, including 1080p (Full HD) and 720p (HD).

Some of the Software components are Arduino IDE, Raspberry pi camera Interface, OpenCV library and Tensorflow.

3. The Arduino IDE

It is an open-source software tool used to write and upload programs to Arduino boards. It enables the development and control of Arduino circuits by facilitating the transfer of written programs to the microcontroller. The IDE utilizes a simplified version of the C programming language for coding Arduino projects. It provides a user-friendly interface for writing, compiling, and uploading code to the Arduino board.

4. Raspberry Pi Camera Interface

Software that allows remote viewing of live images captured bythe Pi camera on a laptop. It provides a platform to view the real-time feed from the Pi camera. Additionally, it offers the capability to record or download images and videos in multiple resolutions, with various customizable settings.

5. OpenCV

A versatile cross-platform library that enables reading and writing of images and videos, as well as processing and transforming images through filters and feature detection. It facilitates the detection of distinct objects in videos or images, such as faces, eyes, and cars. OpenCV is widely used for developing real-time computer vision applications and is focused on video capture and image processing. It supports programming languages like C++, C, Python, and Java, and is compatible with various operating systems including Linux, macOS, Windows, Android, and iOS.

6. TensorFlow

It is a popular open-source Python library that enables users to create deep learning models. With TensorFlow, users can preprocess data, construct models, train them, and estimate their performance. Once the models are built and developed, they can be utilized for various projects and applications.

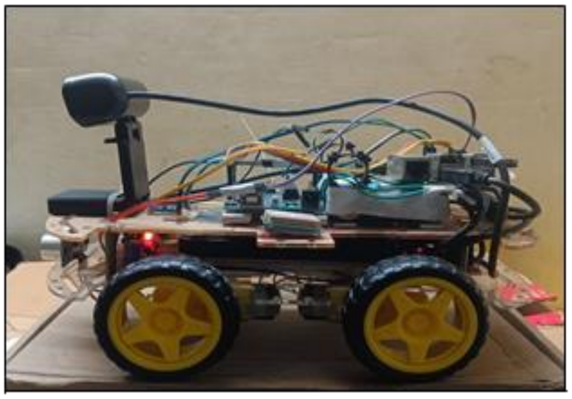


Fig .2. Hardware setup

V.CONVOLUTION NEURAL NETWORK (CNN)

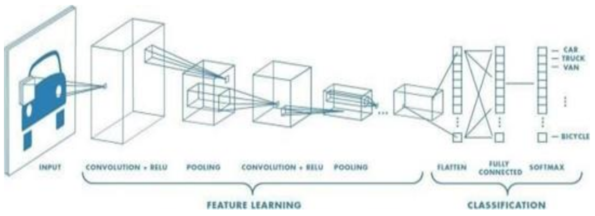
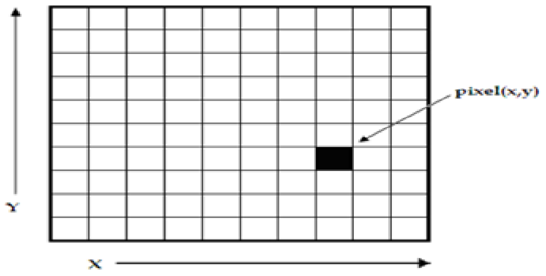


Fig .3.CNN architecture

Object detection plays a crucial role in self-driving cars. Similar to how humans can quickly sense and recognize objects, self-driving cars need to detect and differentiate between various entities like humans, vehicles, traffic lights, and roads. Convolutional Neural Networks (CNNs) are widely employed for image classification in this context. CNNs draw inspiration from the human visual cortex, which is receptive to specific regions of the visual field, enabling effective object detection in self-driving cars.[2]

Fig .4.The input image and its interpretation in pixels

The image represented in "Fig.4" serves as the input for a Convolutional Neural Network (CNN). When the computer processes the image, it interprets it as an array of pixels. After classification, the CNN provides an output indicating the probability of the input image being classified as a specific object. By analyzing edges and curves as low-level features, the computer identifies and recognizes the object depicted in the image. CNNs consist of various layers, including convolutional layers, non-linearity layers, pooling layers, and fully connected layers, which collectively contribute to the network's ability to extract meaningful information from the input image.



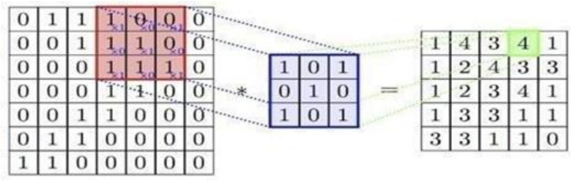


Fig .5. Input image converted into a feature map using a filter.

The initial layer in a Convolutional Neural Network (CNN) is theconvolution layer. In "Fig.3," an image is represented as a 32x32x3 array of pixels. This image is fed into the convolution layer. The convolution layer extracts features by performing mathematical operations between the input matrix and a filter or kernel. The filter, unique to a specific class label, is also a matrix containing weights. Sliding the filters over the input matrix, multiplication and addition of corresponding cells occur, resultingin a feature map as depicted in "Fig.5".To address non-linearity within the neural network, Rectified Linear Unit (ReLU) is utilized.

The dimensionality of the feature map is reduced by pooling technique. The pooling layer extracts the largest element from the feature map by considering 2X2 filters shown in “Fig.6,” thereby retain the original information. Convolutional Neural Networks (CNNs) frequently employ the average pooling technique to downscale or shrink the spatial dimensions of an input feature map. It works by separating the input feature map into rectangular, non-overlapping areas often referred to as pooling windows and replacing each area with the average value of the pixels contained inside. Max pooling can be advantageous in autonomous driving scenarios where the detection of distinct and prominent features is crucial. For example, identifying key objects like traffic signs, pedestrians, or vehicles with high precision can be critical for safe navigation. Max pooling helps emphasize these prominent features and can enhance the network's ability to detect and classify them accurately.

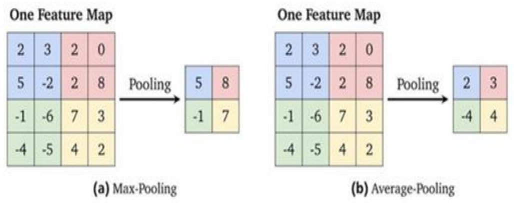


Fig .6.Example for Max pooling of feature map

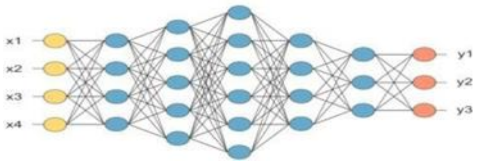


Fig.7.Fully Connected Layer of CNN

The feature map obtained after pooling is fed into the Fully Connected Layer, as shown in "Fig.7," after converting it into a vector. The Fully Connected (FC) Layer is responsible for classifying the image into a specific class. In this layer, the vector is compared with various filters, and multiplication and addition operations are performed between them. By performing these mathematical operations between the vector and different filters, various values are obtained. The maximum value among these values and the filter used for multiplication are identified. The imageis then classified based on the class associated with the identified filter.

Detection Of Objects by self-driving car using YOLO (You OnlyLook Once):

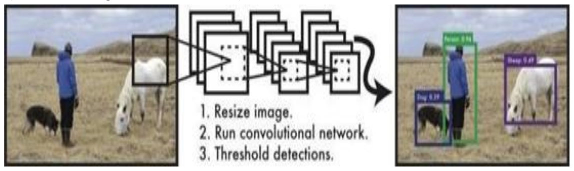


Fig.8. Detecting, classifying and localizing objects in an image byYOLO

YOLO is one among the algorithms that use CNN for object detection. It distinguishes itself by scanning the entire image as a whole during image processing, hence its name. YOLO is known for its high speed, capable of processing images at a rate of 45 frames per second. The neural network is applied to the complete image, enabling it to calculate bounding boxes and class probabilities. In "Fig.9(a)," the entire image serves as input to the CNN. YOLO divides the image into a grid of fixed-size cells, typically 13x13. Each grid cell predicts a fixed number of boundingboxes and their associated class probabilities. Notably, each grid cell predicts only one object.

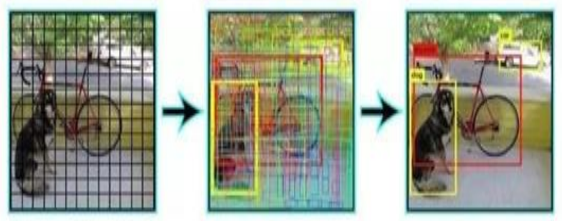


Fig .9(a).Working of YOLO

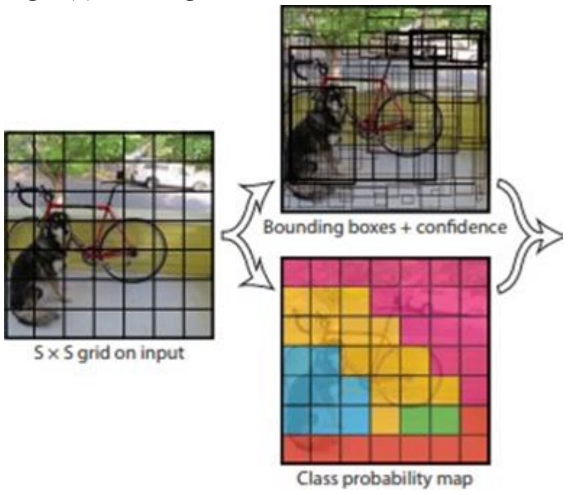


Fig .9(b).Working of YOLO

Each boundary box is associated with five aspects, the confidence score of the box, 'x' and 'y' representing offsets, and 'w' and 'h' representing the width and height of the bounding box, respectively. The confidence score indicates the probability of the box containing an object. The values of 'w' and 'h' are normalized by the width and height of the image, respectively, ensuring that all aspects ('w', 'h', 'x', 'y') are between 0 and 1. Additionally, for each cell in the 13x13 grid shown in "Fig.9(a)," YOLO predicts a fixed number of class probabilities, determining the probability of the object belonging to a specific class.
In "Fig. 9(a)", YOLO predicted a total of 845 bounding boxes within the 13x13 grid. The bounding boxes with probabilities greater than the predefined threshold are considered for object localization. The bounding boxes in detected objects such as a dog, bicycle, and car.
In "Fig. 9(b)", the model divides the image into an S × S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an
$S \times S \times (B * 5 + C)$ tensor.

V.LANE DETECTION

Lane detection plays a critical role in the development of self-driving cars, as it enables the vehicle to perceive and understand the road structure. The process of road lane detection typically involves several key steps to accurately identify and track the lanes.[3][4].

The steps are given below:
Noise Removal:
The first step in lane detection is to preprocess the input video frame by removing noise and artifacts that may interfere with lanedetection algorithms. Common techniques for noise removal include applying filters such as Gaussian blur or median blur to smooth out the image and reduce unwanted details.

Colour Information Discarding:
In lane detection, colour information can sometimes be misleading or inconsistent due to lighting conditions and road variations. To mitigate these issues, the next step is often to convert the input frame to a colour space that is more robust for lane detection. For example, converting the image to grayscale can simplify subsequent processing steps while preserving important lane-related features.

Edge Detection:
Once the image has been preprocessed, the next step is to detect edges that correspond to lane markings. Edge detection algorithms, such as the Canny edge detector, are commonly employed for this purpose. These algorithms identify rapid changes in pixel intensity, highlighting the edges where lane markings are likely to be present.

Region of Interest (ROI):
In a typical road scene, only a specific region is relevant for lane detection. This region is usually defined as the portion of the image where the lanes are expected to appear. By defining a region of interest (ROI), the computational load can be reduced and false detections outside the relevant area can be minimized. The ROI is often defined as a trapezoidal shape at the bottom of the image, covering the expected location of the lanes.

Hough Transform:
The final step in lane detection is applying the Hough transform to identify and extract the lane lines from the edge-detected image. The Hough transform is a popular technique that allows lines to be represented as points in a parameter space, which can then be transformed back into the image space to obtain the line segments corresponding to the lane markings.

Lane Tracking and Stability:
Once the lane lines are detected, additional processing can be applied to track and stabilize the lanes over multiple video frames.This helps to ensure a smoother and more reliable lane detection output. Techniques such as line fitting, interpolation, and averaging can be employed to estimate the position and trajectory of the lanes over time.

Decision-Making and Autonomous Driving:
The lane detection output, along with other information such as vehicle speed, can be used by the decision-making algorithms of the autonomous driving system. The detected lane boundaries provide important cues for vehicle trajectory planning, lane keeping, and decision-making processes. For example, the vehicle can adjust its steering angle and speed based on the detected lane position to stay within the lane and make appropriate driving decisions.

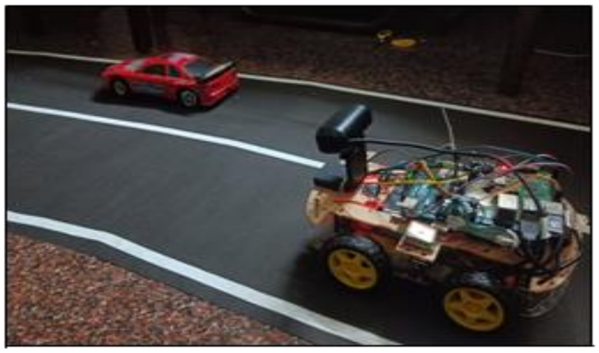


Fig .10.Car following the right path

VI.OBJECT DETECTION

The YOLOv5 model achieved an accuracy of 98% in detecting and classifying pedestrians. This high accuracy level ensured the car's ability to accurately identify pedestrians, a critical factor for ensuring the safety of both the autonomous car and pedestrians sharing the road. The model demonstrated an accuracy of 89.6% in detecting and classifying cars and other vehicles. This accuracylevel was crucial for the car's ability to perceive and respond to other vehicles on the road, enabling safe maneuvering and navigation. The accuracy for detecting and classifying other objects ranged from 65% to 94%, depending on the specific object category

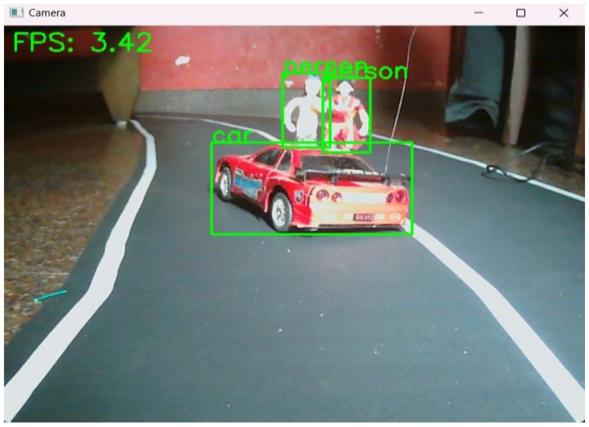


Fig .11.Object detection

VII.POTHOLES DETECTION

The robot effectively detected and responded to road irregularities, adjusting its speed for a smooth and safe navigation experience. This adjustment will prevent discomfort for passengers and minimize the potential damage to the vehicle caused by sudden impacts .Applying the YOLO object identification algorithm, pothole detection uses YOLO seeks out and locates potholes in pictures or video streams. The YOLO model learns the properties and attributes of potholes by being trained on a dataset that comprises annotated instances of them. After being trained, the YOLO model can be used to perform batch or real-time processing inference, locating potholes in fresh, unexplored data. The identified potholes are then refined using post-processing techniques, such as removing false positives and applying non-maximum suppression to get rid of overlapping detections. By superimposing bounding boxes on the original image or video frames, the identified potholes may be seen, allowing for further analysis and decision-making based on their positions and sizes.

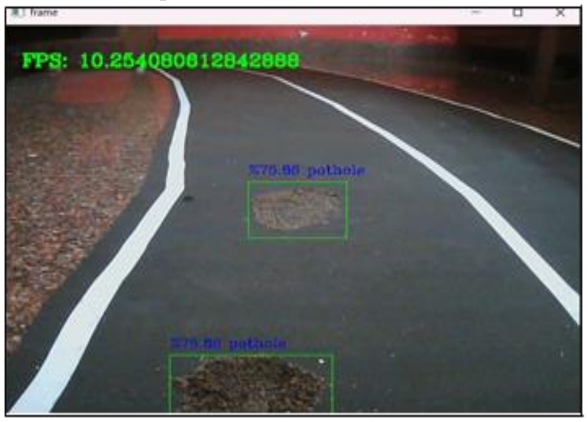


Fig .12.Detection of potholes

VIII.CONCLUSION

In this paper, a prototype of a self-driving car is discussed. Deep learning algorithms YOLO is employed to make immediate decisions for the self-driving car. Detection of objects and lane detection have been studied in this paper. Furthermore, object avoidancea and speed control is discussed.

ACKNOWLEDGMENT

The authors thank and express gratitude towards the support provided by our mentor who guided us in all respects and helped us in achieving the desired outcome.

REFERENCE

[1] Y. A. Badamasi, "The working principle of an Arduino," 2014 11th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2014, pp. 1-4, doi: 10.1109/ICECCO.2014.6997578.

[2] Keiron OShea1 and Ryan Nash2,"An Introduction to Convolutional Neural Networks", Department of Computer Science, Aberystwyth University, Ceredigion, SY23 3DB, School of Computing and Communications, Lancaster University, Lancashire, LA1

[3] H. Lu and J. Yan, "Window frame obstacle edge detection based on improved Canny operator," 2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE), Xiamen, China, 2019, pp.

493-496, doi: 10.1109/EITCE47263.2019.9095074.

[4] Aditya Singh Rathore, "Lane Detection for Autonomous Vehicles using OpenCV Library", B.Tech, J.K. Lakshmipat University

[5] JosephRedmon, Santosh Divvalay, Ross Girshick and Ali Farhadiy, You Only Look Once: Unified, Real-Time Object Detection, University of Washington, Allen Institute for AIy, Facebook AI Research.